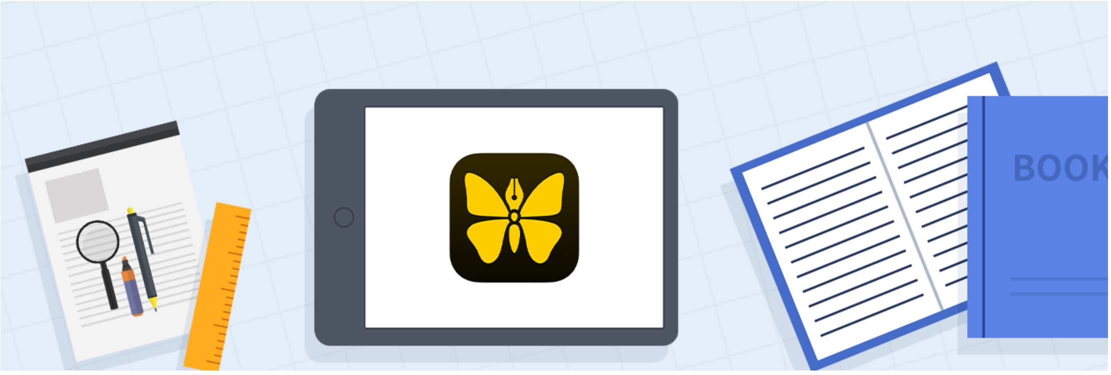


06 月 15 日

第八章：用 Ulysses 搞定学术文档

| 本文为付费栏目文章，您已订阅，可阅读全文 |

如果想在 Ulysses 中完成学术文章写作，除了要用到上一章介绍的 USS（Ulysses Style Sheets），来进行文章的样式排版外，在内容上还必须跨越「三座大山」：**文献引用、交叉引用和数学公式**。

乍看之下，Ulysses 没有提供相对应的功能，对这三类内容无法招架。但是凭借 Markdown 纯文本的便利性，我们可以结合强大的文档转换工具 Pandoc，在 Ulysses 中一口气完成学术文档的写作。

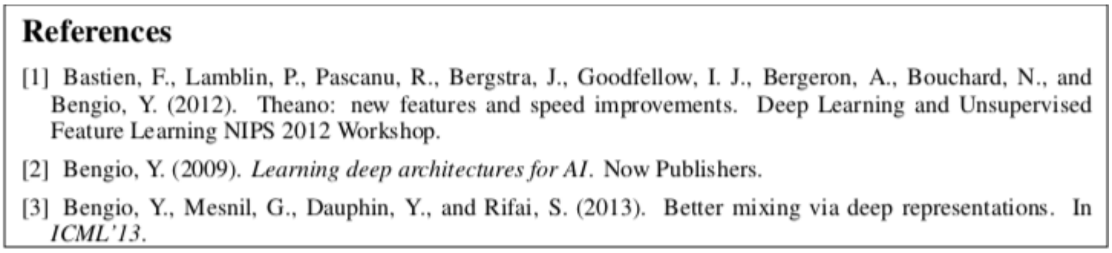
为什么需要 Pandoc

在之前的章节中，我们看到了 Ulysses 在很多场景下，都能独自完成从素材到发布的全过程。但世上没有完美的工具，遇上学术写作「三座大山」，Ulysses 也不得不暴露出了自己的局限性。

学术文章的特别之处

在学术文档中，**对格式的重视程度不低于内容本身**。

虽然办公文档中时常也会给出某些数据的来源，不过通常用脚注的形式标出即可。但在学术写作中，文献必须小心对待。不仅内容上要准确，引用格式也丝毫不能有错。



References

[1] Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I. J., Bergeron, A., Bouchard, N., and Bengio, Y. (2012). Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop.

[2] Bengio, Y. (2009). *Learning deep architectures for AI*. Now Publishers.

[3] Bengio, Y., Mesnil, G., Dauphin, Y., and Rifai, S. (2013). Better mixing via deep representations. In *ICML'13*.

规范的文献引用格式

另一个体现学术文档对格式严格要求的例子是引用，例如图片、公式和表格。以图片为例，在一般文档中，图片下方附有说明即可。但在学术文档中，还必须对图片进行有序编号，且保证文档内所有的图片说明格式统一。

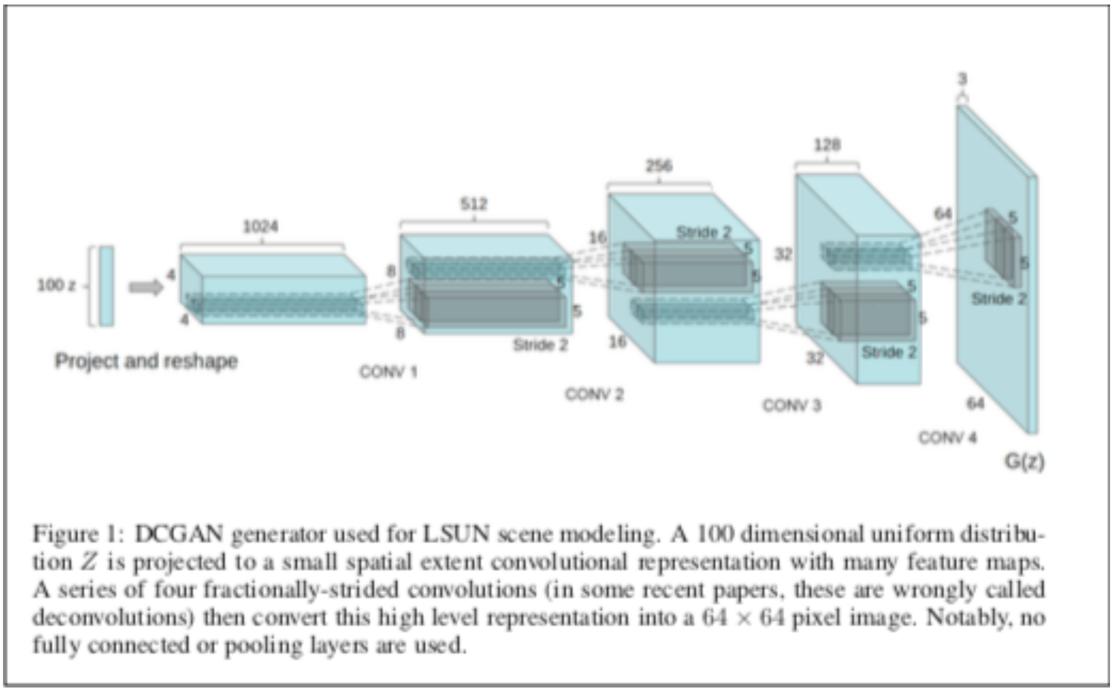

100 z
Project and reshape
1024
4
4
512
8
8
5
5
Stride 2
CONV 1
256
16
16
Stride 2
5
5
CONV 2
128
32
32
Stride 2
5
5
CONV 3
3
64
64
5
5
Stride 2
CONV 4
G(z)

Figure 1: DCGAN generator used for LSUN scene modeling. A 100 dimensional uniform distribution Z is projected to a small spatial extent convolutional representation with many feature maps. A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions) then convert this high level representation into a 64 × 64 pixel image. Notably, no fully connected or pooling layers are used.

图片说明以及编号

Ulysses 遇到了什么困难

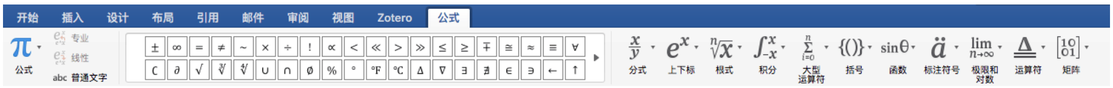
可能你会奇怪，既然 Ulysses 能提供脚注等格式的内容，为什么不能提供插入引用、图片引用的功能呢?

生成引用的难点在于**编号应该随着内容变化**。众所周知，Ulysses 是将 Markdown 的纯文本通过样式文件，变成带格式的富文本。但是在转变的前后，**文本的内容是不变的**，不论是标题、引用或者代码。

但在添加图片说明时，不应该给图片一个指定的编号。否则当我们在第二张和第三张图片中再加入一张时，就必须手动修改后面所有图片的编号。理想情况应该是即使我们中途插入了图片，所有的图片编号应该是自动更新。

而在引用参考文献时，Ulysses 面对的问题是**没法自动生成文献条目的详细内容**。如果你对 BibTeX 或者 Endnotes 等工具熟悉的话，就知道在 LaTeX 或者 Word 中，写作时只需要按类似 `\citep{lamport1986document}` 的格式在文章中写好引用，最后详细的文献条目是可以自动生成，文章中的引用也会变成规范的 `(Lamport, 1986)` 格式。而在 Ulysses 中，就要单独解决文献条目详细信息的保存问题。

至于数学公式，**它的结构相比文本复杂的多**。即使在 Word 中，也需要专门的工具处理，将它结构化成专有的内容格式。而 Markdown 当初是为了网络写作环境而设计，没有相对应的语法标记来实现复杂的数学公式。


开始
插入
设计
布局
引用
邮件
审阅
视图
Zotero
公式
公式
专业
线性
abc 普通文字
分式
上下标
根式
积分
大型运算符
括号
函数
标注符号
极限和对数
运算符
矩阵

Word 中专门的公式工具

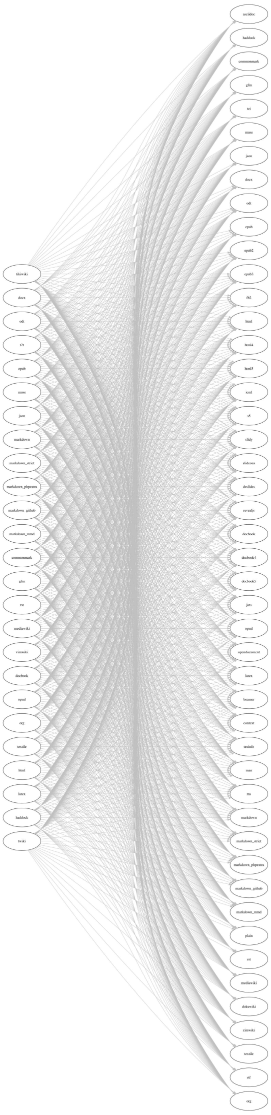
所以想只靠 Ulysses 本身的功能，是没法解决这三类内容的。所幸 Ulysses 本质上仍是一款 Markdown 编辑器，这就能让 Pandoc 助我们一臂之力。

初步了解 Pandoc

Pandoc 并不是一款难上手的工具，但为了你在日后能更加顺手的使用它，而不是仅仅照着给出的代码运行，有必要先了解它的原理和实际意义。

什么是 Pandoc

Pandoc 有很多用法，但归根结底，它是一款**将 Mardkown、LaTeX、DOCX 等多种文件相互转化的文本转换工具**。它到底有多强大呢，官方给出了这张图是最好的回答（如果放大后仍看不清内容，可以点击官网直接查看）。


tikiwiki
docx
odt
t2t
epub
muse
json
markdown
markdown_strict
markdown_phpextra
markdown_github
markdown_mmd
commonmark
gfm
rst
mediawiki
vimwiki
docbook
opml
org
textile
html
latex
haddock
twiki
asciidoc
haddock
commonmark
gfm
tei
muse
json
docx
odt
epub
epub2
epub3
fb2
html
html4
html5
icml
s5
slidy
slideous
dzslides
revealjs
docbook
docbook4
docbook5
jats
opml
opendocument
latex
beamer
context
texinfo
man
ms
markdown
markdown_strict
markdown_phpextra
markdown_github
markdown_mmd
plain
rst
mediawiki
dokuwiki
zimwiki
textile
rtf
org

Pandoc 的转换能力

从图中可以看出来，Pandoc 并不是一款专门为 Ulysses，或者 Markdown 准备的工具。相反是因为 Ulysses 可以方便地导出 Markdown 文件，才使得我们可以利用它。

和我们在 App Store 或者应用官网下载的应用不一样，Pandoc 是一款命令行工具。所谓「命令行工具」，指的是它必须在 macOS 自带的「终端」（或者 iTerm 等应用）中，通过指定的命令运行，而没有图形界面。

在本章中，我会以「终端」应用为例介绍 Pandoc 的使用。如果你对编辑器比较熟悉，可以尝试搜索 Pandoc 的编辑器插件。例如 VS Code 的 vscode-pandoc 就是一款功能比较完善的插件。

安装 Pandoc

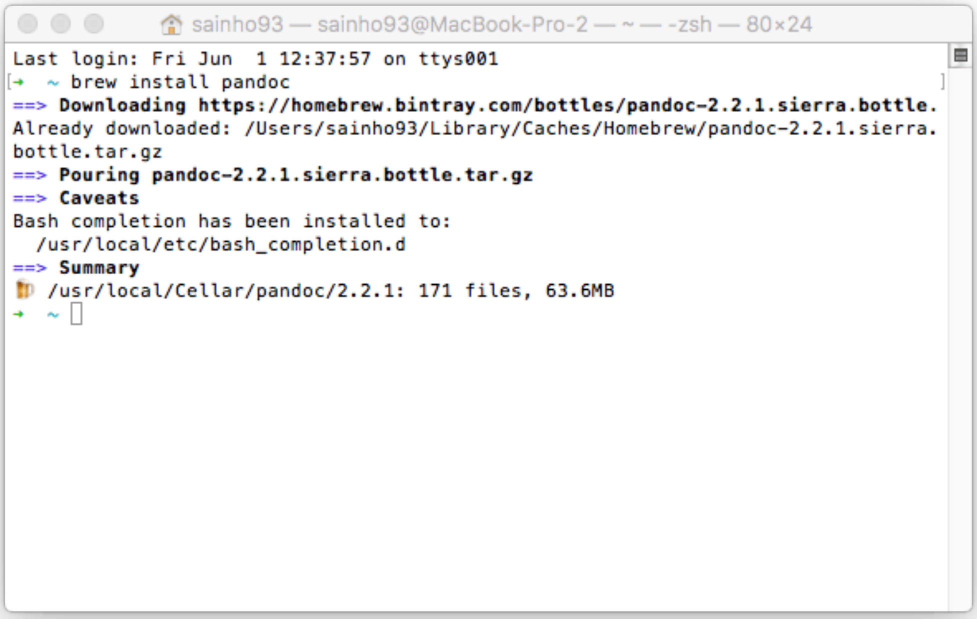
在 Pandoc 官网中，给出了 Pandoc 的安装方法。不过对于这些命令行工具，我建议最好是通过 Homebrew 统一安装和管理。

关于 Homebrew 的介绍和安装，在《像 Mac 高手一样管理应用，从 Homebrew 开始》一文中有详细介绍，这里不作赘述。

为了安装 Pandoc 打开终端，输入命令：

```
brew install pandoc
```

即可安装 pandoc。如果顺利的话，将出现下面的画面。



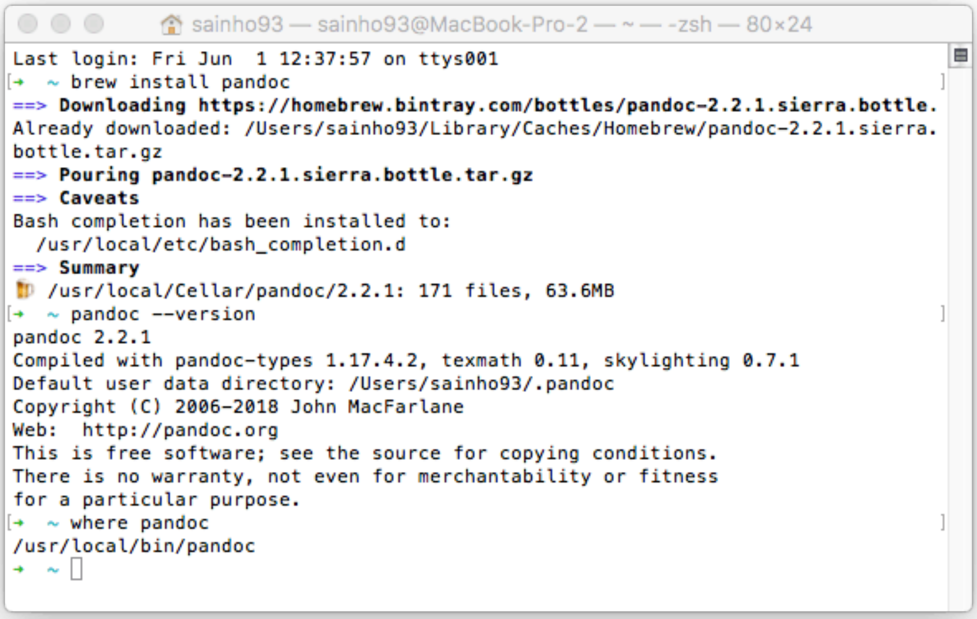
```
Last login: Fri Jun  1 12:37:57 on ttys001
➜  ~ brew install pandoc
==> Downloading https://homebrew.bintray.com/bottles/pandoc-2.2.1.sierra.bottle.
Already downloaded: /Users/sainho93/Library/Caches/Homebrew/pandoc-2.2.1.sierra.
bottle.tar.gz
==> Pouring pandoc-2.2.1.sierra.bottle.tar.gz
==> Caveats
Bash completion has been installed to:
  /usr/local/etc/bash_completion.d
==> Summary
🍺  /usr/local/Cellar/pandoc/2.2.1: 171 files, 63.6MB
➜  ~ 
```

安装 Pandoc 成功

安装之后你可以输入命令

```
pandoc --version
```

来查看自己安装的 pandoc 版本。



```
Last login: Fri Jun  1 12:37:57 on ttys001
→  ~ brew install pandoc
==> Downloading https://homebrew.bintray.com/bottles/pandoc-2.2.1.sierra.bottle.
Already downloaded: /Users/sainho93/Library/Caches/Homebrew/pandoc-2.2.1.sierra.
bottle.tar.gz
==> Pouring pandoc-2.2.1.sierra.bottle.tar.gz
==> Caveats
Bash completion has been installed to:
  /usr/local/etc/bash_completion.d
==> Summary
🍺  /usr/local/Cellar/pandoc/2.2.1: 171 files, 63.6MB
→  ~ pandoc --version
pandoc 2.2.1
Compiled with pandoc-types 1.17.4.2, texmath 0.11, skylighting 0.7.1
Default user data directory: /Users/sainho93/.pandoc
Copyright (C) 2006-2018 John MacFarlane
Web:  http://pandoc.org
This is free software; see the source for copying conditions.
There is no warranty, not even for merchantability or fitness
for a particular purpose.
→  ~ where pandoc
/usr/local/bin/pandoc
→  ~ 
```

查看 Pandoc 的版本

使用 Pandoc 的最佳姿势

Pandoc 使用起来异常灵活，既可以从 Markdown 导出成 DOCX、LaTeX，也可以直接导出成 PDF。那么我们该怎么使用它呢?

其实和办公文档面对的问题类似，在学术文档中，封面、目录、以及各种索引，都是必不可少的。而这些页面更适合在一个可视化的界面中去设计和调整。所以直接从 Markdown 到 PDF 是不太合适的。

而 Pandoc 对 TeX 格式文本非常友好，将 Markdown 文本转成 PDF，其实都是借助 LaTeX 引擎[2]完成的。不过 LaTeX 本身的学习门槛比较高，需要花一定的时间去学习，而且在转化过程中，我们在 Ulyses 中自定义好的样式其实是没法一并转出的。所以这个方案也不合适。

将 Markdown 转成 DOCX，看起来还需要进入 Word 做一些操作，其实是最合适的工作流程。在 Ulysses 中插入参考文献、索引和数学公式，并且利用自定义样式设置好样式。利用 Pandoc，一步就能得到 DOCX 文档。在 Word 中，我们只需要完成封面、目录、附件等内容即可。

Pandoc 与 Ulysses 的协作

看到这里你可能会有些疑惑，既然 Pandoc 只是一款格式转化工具，那么和 Ulysses 内置的导出成 PDF、DOCX 功能有什么不同吗？之前提到的交叉引用、参考文献，还不是没法在 Ulysses 中写作？

接下来，我们就来具体学习，Pandoc 如何帮助我们翻越学术文章的三座大山。

数学公式

对于数学公式，想在 Ulysses 实现类似 Word 中「公式」工具的操作效果是不可能的，因为 Ulysses 说到底还是纯文本的编辑工具。所以可行的做法是，**利用 LaTeX 的数学公式语法来在 Ulysses 中进行写作**，然后利用 Pandoc 将它转换为公式。

Ulysses 作为一款编辑器，实际上你可以在其中写各种格式的标记语言，包括 LaTeX。但问题是，Ulysses 没法解析这些标记语言，所以不能正常预览。而 Pandoc 的出现，不仅是提供了一个转化文件格式的工具，更重要的是，它让 Markdown 文本有了和 LaTeX 「混搭」的可能性。

之前在 Ulysses 中，如果要将文稿转成 DOCX 或者 PDF，所转换的内容必须是 Markdown 格式。而在利用 Pandoc 转换时，Pandoc 可以同时处理文稿中的 Markdown 格式文本和 LaTeX 文本。

所以**无需特别的设置**，在 Ulysses 中，将 LaTeX 格式的数学公式写进一个原始源块中即可。注意两端不要忘了 「$」符号。

```
$\frac {\partial \rho }{\partial t}+\nabla \cdot (\rho \mathbf {v} )=0$
```

之所以要将 LaTeX 内容写进原始源块而不是直接写，是因为要在格式上区分出 Markdown 和 LaTeX，这样 Pandoc 才知道哪部分内容是 Markdown，哪部分是 LaTeX。

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{v}) = 0$$

LaTeX 内容对应的公式

这里提供一些 LaTeX 的辅助资料：

- 如果你对 LaTeX 的数学公式还不太熟悉，可以阅读《一份其实很短的 LaTeX 入门文档》中对数学公式的介绍；
- MilkShake 羊 在《搭建 Ulysses 学术写作之公式输入》一文中，介绍了在 iOS 上使用 MathPad 上手写公式，得到相应的 LaTeX 格式文本的方法；
- 在桌面端，可以使用 HostMath 这个网站上通过点击符号写公式的方式，得到 LaTeX 文本。

在 Ulysses 中写好了公式，剩下的自然是转换成 DOCX 格式。以 Markdown 格式导出文件，保存在桌面。然后打开终端，安装下面的格式输入：

```
pandoc Markdown 文件的文件路径 -o 转换后 DOCX 文件的文件路径
```

例如我想将一份放在桌面的文档进行转换，并将转换后的文档保存在桌面上。

命令就应该是：

```
pandoc /Users/sainho93/Desktop/demo.md -o /Users/sainho93/Desktop/demo.docx
```

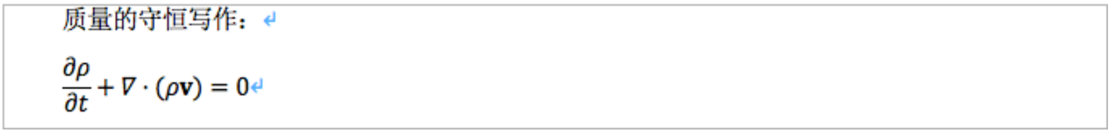
我们来逐个分析使用 Pandoc 的命令。这串命令分为四部分：

1. `pandoc`：首先你需要告诉终端，你要使用哪款工具；
2. `/Users/sainho93/Desktop/demo.md`：这一部分是你想进行转换的文件的路径。其中 `sainho93` 是我的用户名，你应该替换为你 Mac 上的账户用户名。如果不想输入这么长一段内容，可以直接把文件拖进终端，就会自动生成路径；
3. `-o`：Pandoc 的命令符。这一部分是最重要的部分，它指明了你想使用 Pandoc 进行什么操作。在 Pandoc 的官网中，你可以查到所有命令。

这里的 `-o` 是 output 的含义，意思是说执行输出操作。这部分可以同时使用多种命令，在后面我们也会介绍其他命令。

4. `/Users/sainho93/Desktop/demo.docx`：指明转换后的文件路径。注意此时我的桌面上并没有 `demo.docx` 这个文件，这是导出后的文件的文件名。如果你不想输入这么长的内容，可以直接指明文件名 `demo.docx`，Pandoc 就会将它导出到默认目录，即 `/Users/你的用户名` 文件夹中。

在终端中运行命令，如果没有错误提示，应该就能看到 `demo.docx` 文件。打开它就能看到，LaTeX 内容已经被成功转换成公式。在 Word 中点击它，还能继续用「公式」工具编辑。



质量的守恒写作：

$$\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\mathbf{v})=0$$

转换后在 Word 中的效果

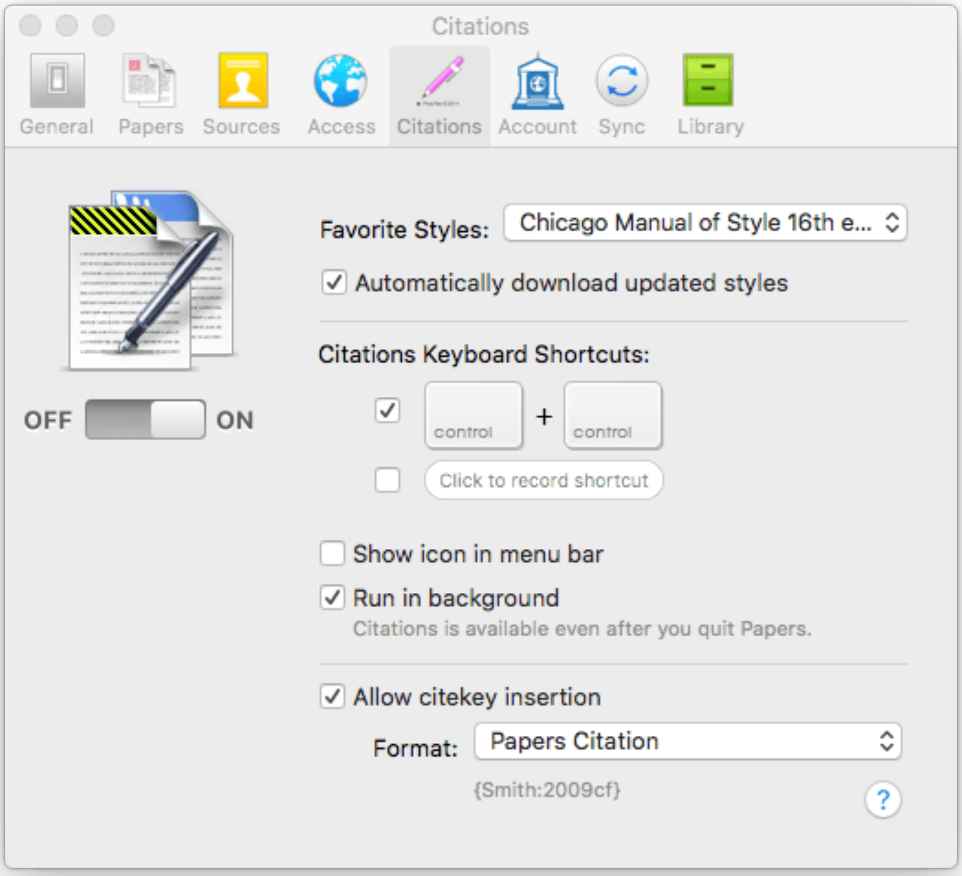
所以，以后你就可以在 Ulysses 中放心的写 LaTeX 公式，再用 Pandoc 转换即可。

参考文献

插入参考文献是学术写作的核心需求，所以已经市面上有许多软件来专门实现它。这里我们介绍两种办法：使用文献管理软件 Papers，以及 Pandoc 的配套工具 pandoc-citeproc。

使用 Papers 插入文献引用

Papers 可以像 iTunes 一样整理你所有的学术文档。当然管理文献的主要目的，还是方便地在写作时插入引用。而它的使用方法也非常简单。


Citations
General
Papers
Sources
Access
Citations
Account
Sync
Library
OFF
ON

Favorite Styles: Chicago Manual of Style 16th e...

☑ Automatically download updated styles

Citations Keyboard Shortcuts:

☑ control + control

☐ Click to record shortcut

☐ Show icon in menu bar

☑ Run in background

Citations is available even after you quit Papers.

☑ Allow citekey insertion

Format: Papers Citation

{Smith:2009cf}

Papers 中的设置

首先在 Papers 偏好设置中**启用引用工具**。然后在 Ulysses 写作时，**呼出引用工具**，在搜索框中搜索文献，点击文献，就可以**在 Ulysses 中插入引用标签**。完成整个文档的写作后，

在 Ulysses 中插入 Citekey

利用之前介绍的办法导出 DOCX 文件（如果没有数学公式，用不上 Pandoc 的话，也可以直接用 Ulysses 自带的导出 DOCX 功能）。在 Word 中，**再次呼出引用工具**，**选择「Format Manuscripts」**，就能看到在文档尾部页面，自动生成了参考文献页面。如果参考文献的格式不满足你的要求，还可以在生成前修改为你想要的格式。

在 Word 中生成参考文献条目

如果你使用的文献管理工具是 Endnote，MilkShake 羊在这篇文章中介绍了类似的办法。

使用 Papers 和 Endnote 插入文献引用的体验都非常流畅，它们展示了一种高效地文献引用流程：

1. 在写作时使用简单的关键词（citekey）作为引用；
2. 根据文档的格式需求设定参考文献的风格；
3. 一键自动生成详细的参考文献条目。

不过作为收费应用，Papers 需要近 70 欧，Endnote 则是 208 欧元，如果你之前是在用 Zotero 等工具来进行文献管理，只为了一个插入文献的功能去购买，显然值得仔细权衡。

而使用 pandoc-citeproc 则是一个完美的免费替代方案。

使用 pandoc-citeproc 插入文献引用

pandoc-citeproc 是专门为了解决插入文献引用而出现的工具，它需要搭配 pandoc 一起使用。和 pandoc 一样，它同样是一款命令行工具，可以在终端中通过 Homebrew 安装：

```
brew install pandoc-citeproc
```

那么 pandoc-citeproc 是怎么工作的呢。

作为一款命令行工具，它实际完成的上述流程中的第二步和第三步，第一步中插入引用关键词，需要我们手动完成。

不过这时候问题来了，我们回想下使用 Papers 的最后一步，其实是将**文章中的引用关键词与 Papers 文献库中的文献元信息对应**了起来，才能在 Word 中一键生成参考文献。而 pandoc-citeproc 本身并不是一款文献管理工具，它要从哪来获得文献的具体信息呢?

为了在使用 pandoc-citeproc 生成网站的参考文献条目，我们必须要先将文献的元信息保存在一个 `.bib` 为后缀名的文件中。`.bib` 其实是 LaTeX 文件生成参考文献是使用的文件，Google Scholar 或者其他文献数据库网站中，可以很方便的获取一篇文献的 BibTeX 格式（`.bib` 文件所需的格式）的文献信息。

获取一篇文章的 BibTeX 内容

获取到文献的元数据后，利用 Textmate 新建一个后缀名为 `bib` 的文件，将搜集到的信息保存在文件中。

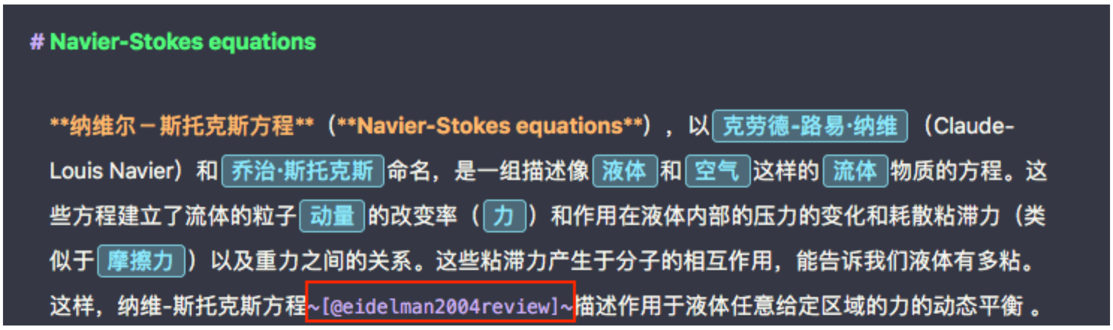
在 `.bib` 文件中存储好文献的信息后，就可以正式开始进行文献引用。在 Papers 中我们没有特别注意引用关键词的写法，因为是 Papers 的引用工具自动生成的。而在使用 pandoc-citeproc 时，则必须小心对待。

文章中的引用部分，应该是「@ + 引用标识符（citation identifier）」的结构。例如对于这篇文献：

```
@article{eidelman2004review,
  title={Review of particle physics},
  author={Eidelman, Simon and Hayes, KG and Olive, KA ea and Aguilar-Benitez, M and Amsler, C and Asner, D and Babu, KS and Barnett, RM and Beringer, J and Burchat, PR and others},
  journal={Physics letters B},
  volume={592},
  number={1-4},
  pages={1--5},
  year={2004},
  publisher={Elsevier}
}
```

为了能正确的区分出不同的引用文献，可以将文献 BitTeX 中括号内的第一行的内容作为引用标识符。再加上 `@` 就组成了文章中引用关键词 `@eidelman2004review`。为了和正文内容区分开来，还要在应用两端加上方括号 `[]`。

我们知道，如果在 Ulysses 中输入方括号，编辑器为认为你要插入链接。为了正常显示，可以将被方括号包裹的引用关键词作为原始源插入。

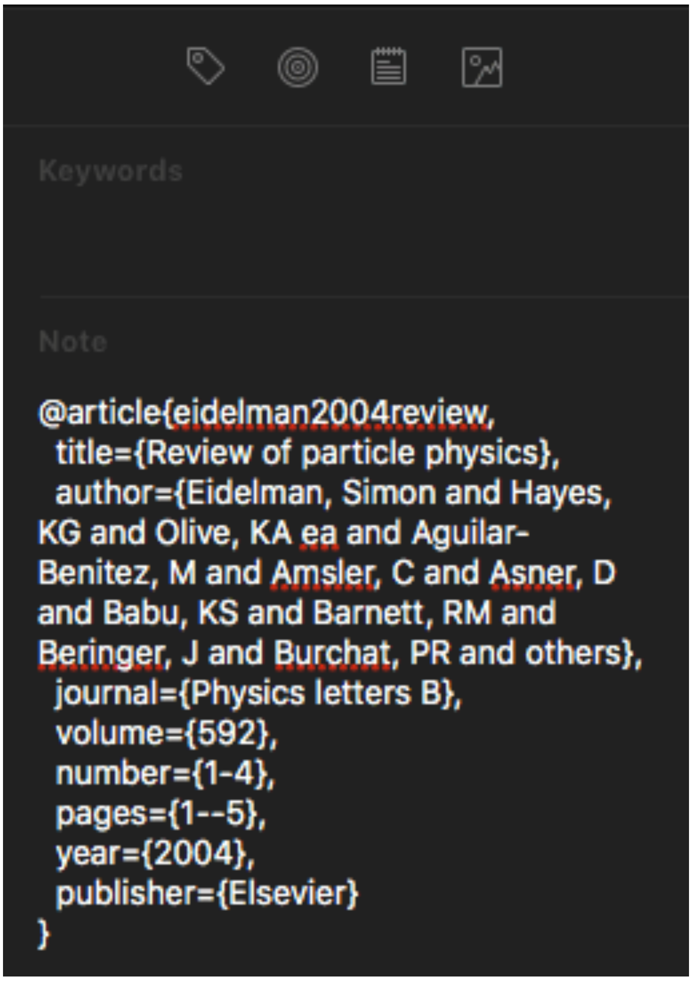


Navier-Stokes equations

纳维尔－斯托克斯方程（**Navier-Stokes equations**），以克劳德-路易·纳维（Claude-Louis Navier）和乔治·斯托克斯命名，是一组描述像液体和空气这样的流体物质的方程。这些方程建立了流体的粒子动量的改变率（力）和作用在液体内部的压力的变化和耗散粘滞力（类似于摩擦力）以及重力之间的关系。这些粘滞力产生于分子的相互作用，能告诉我们液体有多粘。这样，纳维-斯托克斯方程~[@eidelman2004review]~描述作用于液体任意给定区域的力的动态平衡 。

在 Ulysses 中插入引用

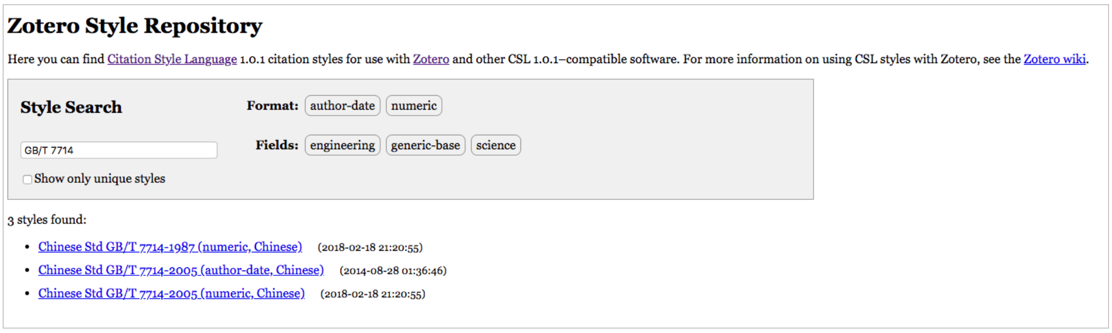
如果在 Mac 上，我们可以一边打开 Ulysses，一遍打开 bib 文件查看不同文献的引用标识符。而在 Ulysses 上要跳出应用去查看就有点麻烦。其实在搜索到文献信息后，可以利用 Ulysses 的附注，将一条文献信息作为一条附注插入。



```
@article{eidelman2004review,
  title={Review of particle physics},
  author={Eidelman, Simon and Hayes, KG and Olive, KA ea and Aguilar-Benitez, M and Amsler, C and Asner, D and Babu, KS and Barnett, RM and Beringer, J and Burchat, PR and others},
  journal={Physics letters B},
  volume={592},
  number={1-4},
  pages={1--5},
  year={2004},
  publisher={Elsevier}
}
```

将文献条目作为附注记录在 Ulysses 中

而**第二步则是根据文档要求的引用格式来设定引用风格**。pandoc-citeproc 默认是 「Chicago Manual of Style author-date format」样式风格，如果学校要求其他的格式，可以在 Zotero Style Repository 中搜索。这里包含几乎所有规范的引用风格，例如国标 GB/T 7714–2005 。搜索到之后下载它的 `.csl` 文件。



Zotero Style Repository

Here you can find Citation Style Language 1.0.1 citation styles for use with Zotero and other CSL 1.0.1–compatible software. For more information on using CSL styles with Zotero, see the Zotero wiki.

Style Search

GB/T 7714

☐ Show only unique styles

3 styles found:

- Chinese Std GB/T 7714-1987 (numeric, Chinese) (2018-02-18 21:20:55)
- Chinese Std GB/T 7714-2005 (author-date, Chinese) (2014-08-28 01:36:46)
- Chinese Std GB/T 7714-2005 (numeric, Chinese) (2018-02-18 21:20:55)

Zotero Style Repository 的搜索界面

做好一切准备工作，就可以开始正式生成参考文献了。在终端中输入：

```
pandoc --filter pandoc-citeproc --bibliography=bib 文件的文件路径 --csl=csl 文件的文件路径 Markdown 文件的文件路径 -o 转换后 DOCX 文件的文件路径
```

例如我将 bib 文件、csl 文件和 Markdown 文件都放在我的用户目录中 `/Users/sainho93`，则命令应该为：

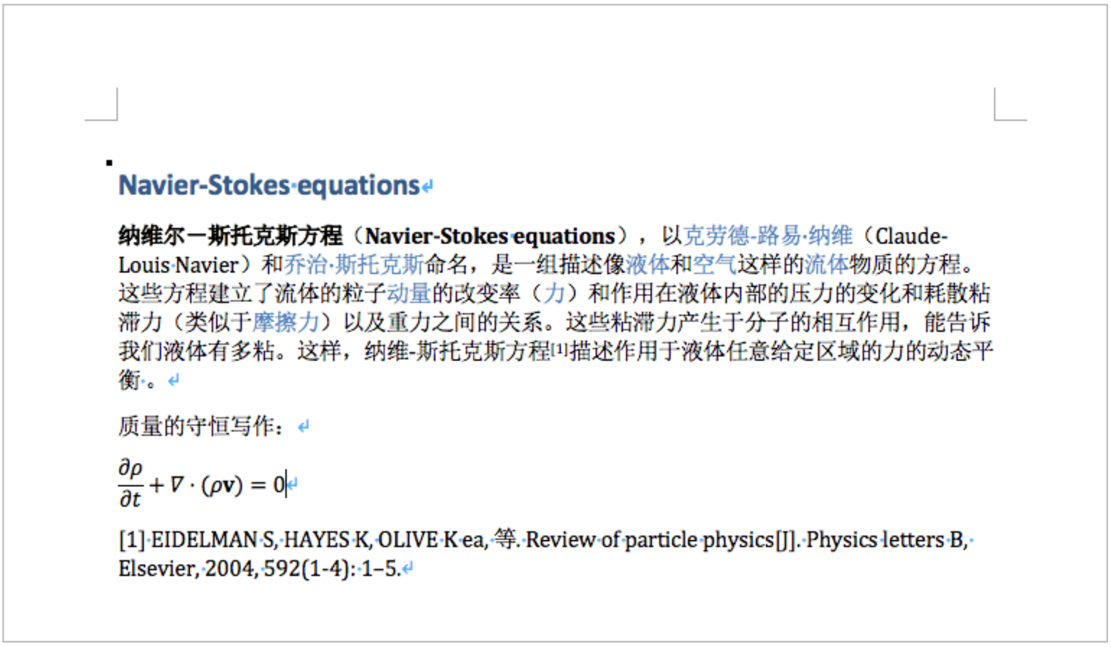
```
pandoc --filter pandoc-citeproc --bibliography=demo.bib --csl=demo.csl demo.md -o demo.docx
```

这次的命令相比上一条，多了三处内容：

- `--filter pandoc-citeproc`：通过 `--filter` 命令，可以将 pandoc-citeproc 与 pandoc 结合起来使用；
- `--bibliography=demo.bib`：通过 `--bibliography` 命令，可以指定参考文献库的文件目录；

- `--csl=demo.csl`：通过 `--csl` 命令，指定参考文献的风格文件的文件目录。

这三条命令都是为了自定义参数而添加的，执行后你就能得到已经生成好参考文献的 DOCX 文件。



Navier-Stokes equations

纳维尔一斯托克斯方程（Navier-Stokes equations），以克劳德-路易·纳维（Claude-Louis Navier）和乔治·斯托克斯命名，是一组描述像液体和空气这样的流体物质的方程。这些方程建立了流体的粒子动量的改变率（力）和作用在液体内部的压力的变化和耗散粘滞力（类似于摩擦力）以及重力之间的关系。这些粘滞力产生于分子的相互作用，能告诉我们液体有多粘。这样，纳维-斯托克斯方程[1]描述作用于液体任意给定区域的力的动态平衡。

质量的守恒写作：

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

[1] EIDELMAN S, HAYES K, OLIVE K ea, 等. Review of particle physics[J]. Physics letters B, Elsevier, 2004, 592(1-4): 1–5.

参考文献在 Word 中的效果

可以看到参考文献被直接附在文章尾部，所以还需要我们稍作处理，在文章和参考文献之间加上分页符和标题等。

交叉引用

搞定了数学公式和参考文献，交叉引用是最后一道难关。

对学术文档不太了解的读者可能不清楚交叉引用的重要性。在文章正文中我们常常会提到一些特定的内容，例如「根据第 X 章的数据」、「根据图 X」、「在表格 X 中可以看到」。如果你在写初稿时将它们的编号「写死」，即写成「第二章」、「看图 3」，之后一旦编号发发生了改变，就必须在全文中小心仔细的去一个个修改。

而交叉引用的作用就在于，给这些经常被提到的内容设定一个定义标签，在写作时不将它们的编号「写死」，而是像**在 Ulysses 中写作时像插入文献引用一样，只需要输入引用标签，然后一次性将它们的编号和格式生成**。

对于参考文献我们使用的是 pandoc-citeproc 工具，而对于交叉引用则是 pandoc-crossref。同样是在终端中通过 Homebrew 插入：

```
brew install pandoc-crossref
```

经过了对 Pandoc-citeproc 的了解，相信对 Pandoc-crossref 也能很快的熟悉起来。

设定定义标签

使用 pandoc-crossref 创建交叉引用分为两步。首先是**对被引用的对象设定定义标签**。定义标签首先要用 `#` 指明标示内容的类别。你可以对：

- 图片 – fig
- 表格 – tbl
- 公式 – eq
- 章节 – sec
- 代码块 – lst

这五类内容设定索引。然后对于某个特定的内容，自己定义一串字符，设定为这部分内容唯一的定义标签。定义标签用 `#` 引导，用 `{}` 包裹起来，直接跟在希望引用的内容后面。例如对一个图片我们可以设定为：

```
(img){#fig:pandoc}
```

在文中插入引用标签

而第二步则是在文中，采用和插入引用的类似的格式，在合适的位置写引用标签。引用标签的格式和参考文献的引用类似，也是用 `@` 和 `[]` 来包裹引用内容，和正文进行区分。例如：

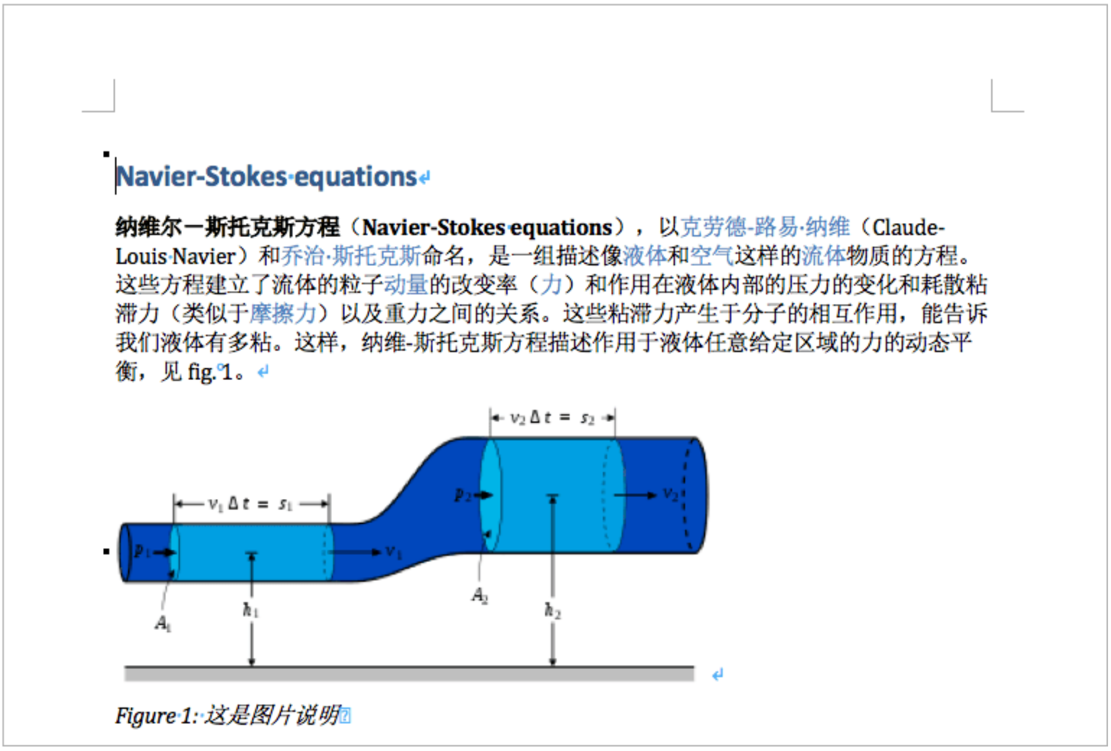
```
看下图~[@fig:pandoc]~可以看出 Pandoc 强大的转换能力。
```

然后就可以利用 pandoc-crossref 类似进行转换：

```
pandoc -filter pandoc-crossref Markdown 文件的文件路径 -o 转换后 DOCX 文件的文件路径
```

和之前的相比，只是将 pandoc-citeproc 换成了 pandoc-crossref， 可见

两种工具的使用方式是一样的。



Navier-Stokes equations

纳维尔一斯托克斯方程（Navier-Stokes equations），以克劳德-路易·纳维（Claude-Louis Navier）和乔治·斯托克斯命名，是一组描述像液体和空气这样的流体物质的方程。这些方程建立了流体的粒子动量的改变率（力）和作用在液体内部的压力的变化和耗散粘滞力（类似于摩擦力）以及重力之间的关系。这些粘滞力产生于分子的相互作用，能告诉我们液体有多粘。这样，纳维-斯托克斯方程描述作用于液体任意给定区域的力的动态平衡，见 fig. 1。

Figure 1: 这是图片说明

转换后的图片引用

除了图片，当然还可以对其他类型内容进行索引：

```
//对数学公式设定定义标签
$$ math $$ {#eq:label}


//对表格设定定义标签
```

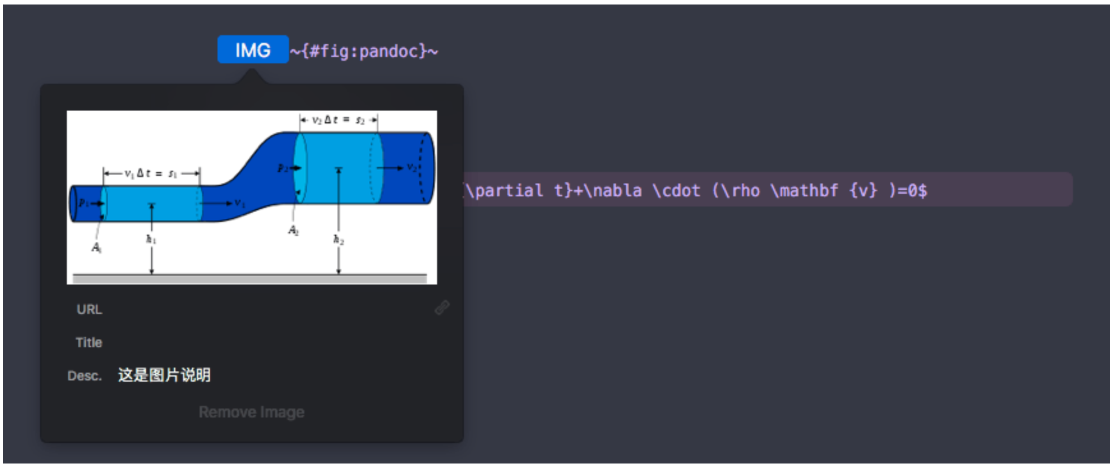
```
a   b   c
--- --- ---
1   2   3
4   5   6

: Caption {#tbl:label}

//对章节名设定定义标签
Section {#sec:section}
```

打开转换后的，可以看到不止图片被编号，之前图片引用中的说明，也自动变成了图片说明。

在使用图片的交叉引用时，对于图片说要额外注意。在 Ulysses 中插入图片时，**一定给图片加上图片说明**，否则 pandoc-crossref 会不能正常运行。


IMG
URL
Title
Remove Image
$v_2 \Delta t = s_2$
$v_1 \Delta t = s_1$
p_1
p_2
v_1
v_2
A_1
A_2
h_1
h_2

```
~{#fig:pandoc}~
\partial t}+\nabla \cdot (\rho \mathbf {v} )=0$
```

Desc. 这是图片说明

图片说明

而用 Ulysses 导出图片时，图片文件和 Markdown 文件会被放在同一个文件夹中。但是在 Markdown 文稿中以标准的插入图片语法标记`![]()`呈现时，括

号中的图片路径只会写成图片的名称。这会造成 Pandoc-crossref 不能正确的找到图片资源，也就无法成功转换。

为了解决这个不完美之处，你可以在导出 Markdown 文件后，暂时将图片放到 `/Users/你的用户名/`目录下。或者在 Ulysses 中写作时，就直接使用标准的插入图片语法标记，用原始源将语法标记和定义标签一起包裹起来，例如：

~   {#fig:pandoc}~。

修改交叉引用的语言

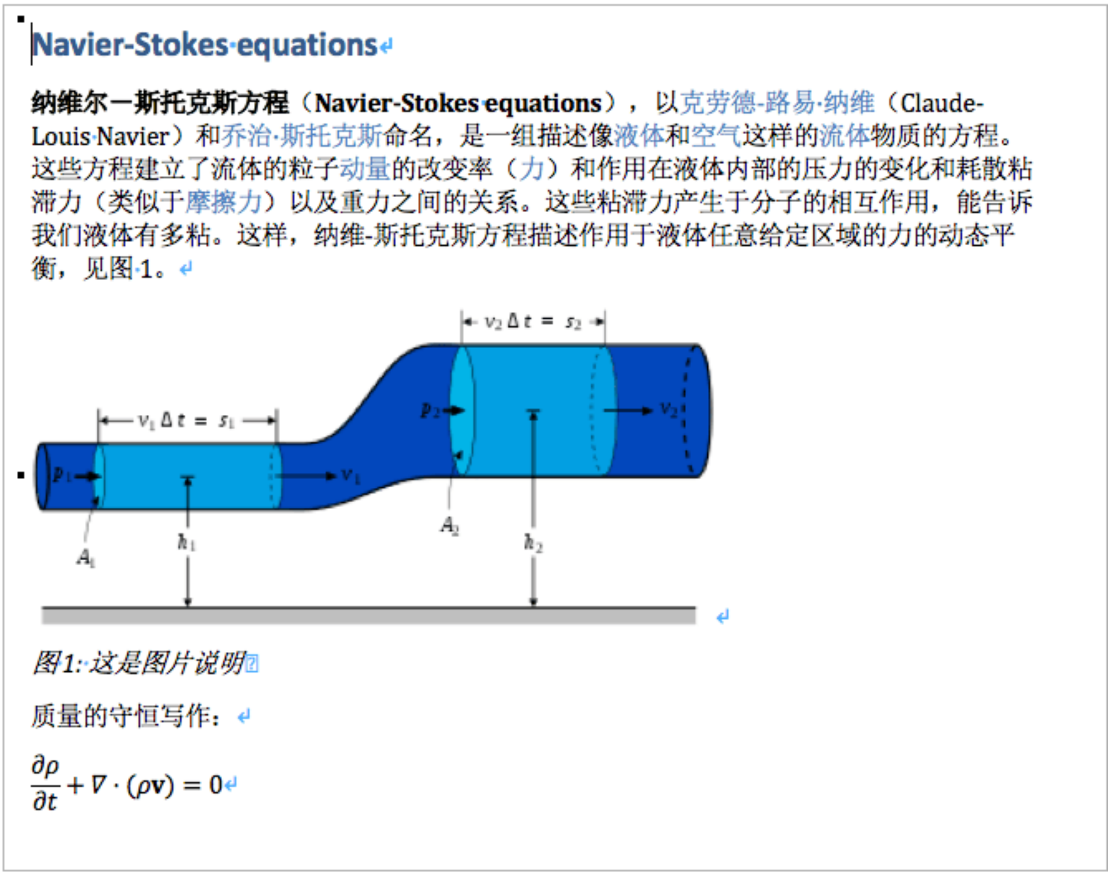
不过也可以看到，无论是文中的引用部分，还是图片编号，都是默认的英文。

文中的引用部分非常好修改，只要在插入时，写成`[图 @fig:pandoc]`，导出后的文档中的「fig」就会自动变成「图」。当然你会可以根据自己的需求设定不同的前缀。

而图片说明中的语言修改起来稍微复杂一些，需要在文档的开头添加下面内容：

```
---
figureTitle: "图"
...
```

重新运行一遍转换命令，就能看到新生成文档中，原来的英文都被修改成中文。



Navier-Stokes equations

纳维尔—斯托克斯方程（Navier-Stokes equations），以克劳德-路易·纳维（Claude-Louis Navier）和乔治·斯托克斯命名，是一组描述像液体和空气这样的流体物质的方程。这些方程建立了流体的粒子动量的改变率（力）和作用在液体内部的压力的变化和耗散粘滞力（类似于摩擦力）以及重力之间的关系。这些粘滞力产生于分子的相互作用，能告诉我们液体有多粘。这样，纳维-斯托克斯方程描述作用于液体任意给定区域的力的动态平衡，见图1。

图1: 这是图片说明

质量的守恒写作：

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

转换语言后的效果

使用 Ulysses 中的样式

翻越了数学公式、交叉引用和参考文献的三座大山，我们基本已经靠 Pandoc 解决了学术写作时的内容需求。不过使用 Pandoc 也带来一个「副作用」：

Pandoc 把文章从 Ulysses 中移出来，导致在 **Ulysses 创建好的自定义样式没法使用**了。

不过这个问题其实靠 Pandoc 自己就能解决。在将 Markdown 转换成 DOCX 的过程中，Pandoc 其实是一直靠内置的一份模版，来设定转换后生成的 DOCX 文件的样式。而在终端中执行命令时，可以加上 `--reference-doc` 命令，来制定一份样式模版。那这个自定义的模版该从哪获得呢?

其实最好的模版，就是我们要转换的文档本身。利用 Ulysses 导出 DOCX 时，可以随意选择我们设定好的自定义样式。这样不仅让之前设定好的样式派上了用场，又不用重新在 Word 中去设置样式。可谓完美地解决了 Pandoc 带来的问题。

所以最合理的操作流程是：

1. 在 Ulysses 中完成写作，包括引用、公式和参考文献；
2. 直接从 Ulysses 中到处一份 DOCX 文件；

3. 利用 Pandoc、pandoc-crossref、pandoc-citeproc 等工具进行转换，将样式模版设定为上一步中到处的 DOCX 文件。

将文档导出成 DOCX 后，可以改名为 `custom-reference.docx`，方便与正式文档进行区分。然后在终端中输入：

```
pandoc --reference-doc=custom-reference.docx Markdown 文件的文件路径 -o 转换后 DOCX 文件的文件路径
```

就能看到导出后的文档， 保持了和从 Ulysses 中直接导出 DOCX 一样的样式。

小结

看完了 Pandoc 与 Ulysses 的结合使用，我们再来梳理一遍使用 Ulysses 完成学术写作的流程。

在初稿阶段，可以完全在 Ulysses 中写作。即使是数学公式、交叉引用、参考文献等内容，可以先写上 LaTeX 和引用标签。初稿完成后，就可以在 Ulysses 中自定义样式。随后就可以利用 Pandoc，将文稿导出成 DOCX，并且自动生成数学公式 、交叉引用、参考文献等内容。最后再在 Word 对排版进行微调，加上封面、目录等内容即可。

至此，我们就完成了 Ulysses 相关所有应用场景的学习。

纵观全教程，Ulysses 最核心的还是它作为写作工具的体验。但是借助 CSS 与 USS，它能在不同的场景中，完成相当一大部分的排版任务，这让它有了发挥超越价值的机会。

希望它能在你们手中，都发挥无可限量地价值!

1. DOCX 文件实际上依然是一种自定义格式的 XML 文件。↩
2. 可以选择其他工具作为中转，不过利用 LaTeX 是功能最全的方案。↩